EPFL

POCS Recitation: RON + Content Distribution

Katerina Argyraki

School of Computer & Communication Sciences

Paper style

- Proposal after Internet architecture and TCP/IP stack had been set
- Much more stylized than BL Name Service
- Evaluation
- Authors had to make compelling case for changing things
- A big challenge with networking research today

5 basic questions

- Problem
- Goals
- Proposed solution
- Evaluation
- Results

Problem

When a source cannot reach a destination because of a link failure,
the Internet routing system may not detect the failure
or take tens of minutes to detect and overcome it.

Main goal

- Consider a set of communicating nodes
- As long as there exists a functional path from one node to another, find it within seconds

- *What is the abstraction that the Internet network layer exposes to Internet end-systems?*
- *What is the abstraction RON exposes to its clients?*

Other goals

- Application-specific criteria for path selection
- A way to express and apply policies
  (which traffic should be allowed to use each path)

Solution

- Nodes form an overlay
- Each node monitors the path to
  and exchanges information with every other node
- Each node may route to another node directly
  or through an intermediate node
- Insight: there typically exist many paths between two nodes;
  two paths are unlikely to share he same problematic link

Evaluation

- Implemented two RONs with nodes located across the US and Europe
- Measured the quality of “direct” and “indirect” paths between pairs of nodes
- Assessed whether indirect paths improved connectivity between nodes

Main results

- It works: RONs detected and bypassed most outages between nodes
- In certain cases, RONs improved communication quality between nodes
- One extra hop is enough

Revisit problem

- When a source cannot reach a destination because of a link failure, the Internet routing system may not detect the failure or take tens of minutes to detect it.

- "Routing scalability comes at the cost of fault tolerance"

Routing (BGP) basics

- Internet divided in Autonomous Systems (ASes)
- Each AS owns a set of IP prefixes
- Each border router announces which IP prefixes it wants to receive traffic for
- It's a path-vector (not a link-state) protocol:
  route announcement propagates hop by hop
- If a router loses "BGP session" with peer (neighbor),
  it withdraws all the routes announced by that peer

Scalability

- (Informally) As system grows, it maintains its properties (including performance) at a reasonable cost per component
- Cost = memory, CPU, bandwidth

- *How does BGP achieve scalability at the cost of fault tolerance?*

Route aggregation

- Many IP subnets represented with the same IP prefix
- Helps scalability: routers keep forwarding state per IP prefix, not IP subnet
- Affects fault tolerance: if the route to an IP subnet (but not the IP prefix) fails, there may exist an alternative route that remains undetected/unused

Parenthesis

- How much memory would a router need to store forwarding table entries to all IP subnets?
- Assume 4 billion IP subnets (1 per IP address)
- Assume 10 bytes per entry -> 40 billion bytes = 40 GB
- Is that so much? How much DRAM does your laptop have?
- Actually, it is. Routers do not typically have separate DRAM chips. They have on-chip SRAM. Takes a lot of space => only tens of MB.

Delayed updates

- Path-vector (not link-state) protocol
- Route announcement (including withdrawal) happens hop by hop
- Administrators limit the announcement rate
- Helps scalability: limits CPU and bandwidth
  devoted to processing announcements
- Affects fault tolerance: it may take minutes for a withdrawal
  to reach the other end of the Internet

Coarse failure detection

- If a router loses BGP session with peer,
  it withdraws all the routes announced by that peer
- Helps scalability: each router monitors only the liveness of BGP sessions
- Affects fault tolerance: route quality may degrade
  even if BGP session is alive

Revisit approach

- *How exactly does RON fix BGP problems?*
- Route aggregation: potentially a different route between each pair of nodes
- Delayed updates: link-state routing
- Coarse failure detection: traffic monitoring

- Routing without any of the BGP scalability mechanisms

A systems paper

- The Internet routing system could not do these things that RON does
- The Internet network layer cannot detect and route around outages within seconds
- The paper's argument is that this problem needs to be solved
  above the network layer, because that's how we can make it scale

Evaluation

- *What experiment does one want to see after understanding the problem and proposed solution? Why would one question whether it would work?*
- It could be that most outages happen at the network edge, behind the RON nodes, in which case RON would not make sense…
- The paper provides experimental evidence that that is not the case

Main goal

- Consider a set of communicating nodes
- As long as there exists a functional path from one node to another, find it within seconds

- *What is the abstraction that the Internet network layer exposes to Internet end-systems?*
- *What is the abstraction that RON exposes to its clients?*

Parting thought

- Internet network layer: as long as there exists a functional path between two nodes, system does its best to find it
- RON: as long as there exists a functional path between two nodes, system does its best to find it within seconds
- “Small” difference in abstraction changes completely the way the abstraction is implemented and at which layer it can be provided

Scaling content distribution

- How can a content provider serve more clients
  without increasing the transmission rate of its link to the Internet?

- Partly solved through transparent caches
- Partly solved through Content Distribution Networks + dynamic DNS

- Reduce user-content distance
- Distribute load across content servers

These are “dirty” solutions

- Transparent caches hijack TCP connections
- CDNs duplicate network-layer functionality
- Both still need extra round-trip for DNS

- Layer violation or duplication + still limited by DNS

(TRIAD project) Clean solution: a new layer

- New content layer between transport and network
- Network layer: finds path between two end-systems
- Content layer: finds path between an end-system and a piece of content
- Content-layer header specifies target content by name
- Content-layer switches forward packets based on content name
- Content-layer switches may cache content and respond to client

Clean solution: a new layer

- Integrates DNS into the TCP/IP stack
- Subsumes transparent caches and CDNs
- Removes DNS round-trip
- Changes the layering to avoid violation and duplication

Clean solution: a new layer

- Forwarding based on content names
- They could be hierarchical, but could they enable efficient aggregation?

Deployment

- Not all switches need to be content-layer switches
  - *the same way not all switches are network-layer switches*
- Only the switches (routers) at domain boundaries
  - *the same ones that run e-BGP*
- Content-layer switches form an overlay over network-layer switches

IP addresses = transient routing tags

- Used for routing within each domain
- Do not need to be global

- Internet = network of NAT domains

(Also resolves source address spoofing)

- True domain-level path encoded in header
- Source cannot hide the origin domain
- Can spoof only its local neighbors

- Restricts spoofing within local domain

(Also resolves reflector attacks)

- True domain-level path encoded in header
- Source cannot hide the origin domain
- Can spoof only its local neighbors

- Restricts reflection within local domain

Scaling content distribution

- New content layer

- Content names = globally unique
- IP addresses = unique within their domain

- Content switches forward by content name and cache content

- Scalable content distribution
- No need for IPv6
- Limited spoofing and reflector attacks